Int J Syst Assur Eng Manag (February 2023) 14(1):429–447
https://doi.org/10.1007/s13198-022-01813-z

ORIGINAL ARTICLE

Effective time context based collaborative filtering recommender system inspired by Gower's coefficient

Gourav Jain[1] · **Tripti Mahara**[2] · **S. C.Sharma**[1]

Received: 29 November 2021 / Revised: 11 May 2022 / Accepted: 24 November 2022 / Published online: 25 December 2022
© The Author(s) under exclusive licence to The Society for Reliability Engineering, Quality and Operations Management (SREQOM), India and The Division of Operation and Maintenance, Lulea University of Technology, Sweden 2022

Abstract The fast growth of Internet technology in recent times has led to a surge in the number of users and amount of information generated. This substantially contributes to the popularity of recommendation systems (RS), which provides personalized recommendations to users based on their interests. A RS assists the user in the decision-making process by suggesting a suitable product from various alternatives. The collaborative filtering (CF) technique of RS is the most prevalent because of its high accuracy in predicting users' interests. The efficacy of this technique mainly depends on the similarity calculation, determined by a similarity measure. However, the traditional and previously developed similarity measures in CF techniques are not able to adequately reveal the change in users' interests; therefore, an efficient measure considering time into context is proposed in this paper. The proposed method and the existing approaches are compared on the MovieLens-100k dataset, showing that the proposed method is more efficient than the comparable methods. Besides this, most of the CF approaches only focus on the historical preference of the users, but in real life, the people's preferences also change over time. Therefore, a time-based recommendation system using the proposed method is also developed in this paper. We implemented various time decay functions, i.e., exponential, convex, linear, power, etc., at various levels of the recommendation process, i.e., similarity computation, rating matrix, and prediction level. Experimental results over three real datasets (MovieLens-100k, Epinions, and Amazon Magazine Subscription) suggest that the power decay function outperforms other existing techniques when applied at the rating matrix level.

Keywords Collaborative filtering · Recommendation system · Similarity measure · Time decay function · Gower's coefficient

✉ Gourav Jain
jaingourav3010@gmail.com

Tripti Mahara
triptimahara@gmail.com

S. C.Sharma
subhash.sharma@pt.iitr.ac.in

[1] Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India

[2] Christ University, Bengaluru, India

1 Introduction

The Internet has become an integral part of people's lives since its inception in the twentieth century. There is a continuous data explosion with the advent of Web 2.0, social media, and online social networking. According to the report, 2.5 quintillion data bytes was generated per day in 2020, and by 2025, this number will increase to 463 exabytes of data (Bulau 2021). As a result of this data explosion, the user's ability for data absorption has reached its limit, and the issue of adverse selection has risen. Finding the right information from this data is like searching for a needle in the haystack. In this, the recommendation system (RS) (Jain et al. 2020) plays an important role. It is a software technology that serves as an information filtering tool to recommend the most favourable items to users depending on their previous personal preferences. The popularity of the recommendation system is continuously rising and is deployed in many domains like music, movie, news, joke, health care, article recommendations, etc. (Konstan et al. 1997; Jain et al. 2013; Anandhan et al. 2018). On e-commerce platforms (Jin et al. 2020), RS helps the users by suggesting the items of their interest. The presence of the Long Tail phenomenon

(Suryakant and Mahara 2016) is also one of the prominent reasons for the ever-increasing popularity of the RS. According to this, users can find popular products quickly, but products in the Long Tail are more difficult and time taking to find. An RS solves this problem by recommending all related items, even if they aren't very popular.

Generally, there are three kinds of recommendation systems: (1) Content-Based (CB), (2) Collaborative Filtering (CF), and (3) Hybrid recommendation. In a content-based strategy (Wang et al. 2017), sufficient information about users and items is required to develop the profiles and provide recommendations. This method recommends the best-matched item after examining the previously rated items. The CB technique can change its recommendations very quickly according to the changes in the user's preferences, but enough information about users and items is required to create profiles. Unlike the CB technique, CF (Schafer et al. 2007; Jain and Mahara 2019) technique considers only the user-item ratings. It forecasts the utility of items for the target user based on the items previously rated by other users. The main benefit of CF is that, it requires less information about users/items to construct profiles and is more accurate than content-based techniques. It is divided into Model-based (Isinkaye et al. 2015) and Memory-based methods (Ghazarian and Nematbakhsh 2015). In the model-based approach, partial ratings are used to train the model, and once the model is trained, it is used to generate quick predictions. The memory-based method predicts the missing ratings based on the evaluations from other users/items. In this method, it is important to select a suitable similarity measure as it helps in finding similar users/items. Many experimental results show that the memory-based method has practical advantages such as simplicity, efficiency, and accuracy. Memory-based methods are classified into User-based (Tan and He 2017) or Item-based (Kant and Mahara 2018) techniques. When the similarity is calculated among users, it is called the user-based CF method; otherwise, it is known as item-based CF. After calculating the similarity among users/items, neighbors are determined for the target users to predict their unknown ratings. The hybrid methods (Ghazanfar and Prugel-Bennett 2010; Wang et al. 2017) combine collaborative and content-based approaches. Most of these similarity measures in the memory-based approach suffer from the data sparsity problem (Yu and Huang 2017; Kant and Mahara 2018), which occurs when the ratio of ratings needed to be predicted to the ratings already available is very high. For instance, predicting the 85% ratings from 15% of the available data. This problem gets aggravated with a continuous increase in users and items. A new similarity measure iGJ is proposed in this paper to overcome this. We compared the new method's performance with the existing method, and the experimental results show that our proposed method is superior.

In addition, the time-aware recommendation systems have been widely researched in recent years, and they have been found to be more successful than standard non-relevant recommendation systems (He and Wu 2009; Campos et al. 2014). Exploiting the context (e.g., location, time, weather, device, and mood) in which users express their preferences have been demonstrated to be very effective in increasing the performance of the recommendation system (Adomavicius et al. 2011). Time-aware recommendation (Ding and Li 2005; Koren 2009) systems focus on the idea that users' attraction to items in online systems diminishes over time. It means that users' most recent ratings on items reflect their current trend on such items. Although there has been a lot of research in this area, only a few studies described how time-based functions could help increase the recommendation system's performance (Larrain et al. 2015). Most research does not precisely describe which time functions should be used and when they can be integrated. This research aims at addressing this gap by applying various time decay functions at three levels, including rating matrix level, similarity computation level, and prediction level, during the recommendation process. The main contributions of this research are as follows:

1. A novel CF-based RS algorithm is proposed to tackle the sparsity issue. The algorithm is validated by applying it to real-world data sets. The results confirm that the method is effective and scalable and outperforms existing CF-based methods.
2. The concept of time was not considered in the traditional CF algorithms, but it is important as the user's preferences change with time. Therefore, various time decay functions are integrated at three levels in the recommendation process to incorporate the time aspect. The experimental results on three real datasets indicate that the proposed time-based method is superior to all other methods.

The outline of the paper is as follows. The related work on CF and time context functions is presented in Sect. 2. The proposed algorithm is described in Sect. 3. The experiments and performance analysis are presented in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Related work

The Collaborative Filtering (CF) (Al-bashiri et al. 2017) is a popular suggestion approach that has been used on a variety of e-commerce platforms. It suggests potential items for target users by automatically learning and analysing their past preferences. In recommendation system, $U = \{U_1, U_2, \ldots, U_M\}$ and $I = \{I_1, I_2, \ldots, I_N\}$ be a set of users and items respectively, and all user rating data are regarded as a user–item rating matrix $[r_{ui}]^{M\times N}$. In this matrix, M and N represent the number of users and items, respectively; r_{ui} is a rating value made by the uth user on the ith item. This rating matrix is sparse in general, which means a substantial number of user's ratings are unknown. As a result, a significant research emphasis of memory-based CF is how to construct an effective similarity measure to deal with the data sparsity problem. The literature about some traditional and recent work on similarity measures is discussed in Sect. 2.1. Many time decay functions are used to analyze the system performance, and a brief discussion on them is given in Sect. 2.2.

2.1 Collaborative filtering-based recommendation

The Collaborative Filtering is one of the most popular methods of RS, which takes the users' preferences for items stored in a database (user-item matrix). It then makes recommendations based on the similarities calculated by a similarity measure. A similarity measure is a statistical measure used to show how two users or items are related. The CF technique is developed on the premise that users with common interests in the past will also share similar tastes in the future. Data sparsity and Cold-start are two main challenges faced by any CF-based RS (Patra et al. 2014). To alleviate them, many similarity measures, i.e., Cosine, PCC, TMJ, Rating Jaccard, RJaccard RJMSD, IPWR Var, Rating Jaccard RPB, etc., have been introduced in recent years. Table 1 highlights some of the traditional similarity measures along with newly developed methods.

Cosine (COS) (Su and Khoshgoftaar 2009) is a conventional similarity measure that computes similarity by calculating the cosine angle formed between user rating vectors. The main drawbacks of cosine measure are (1) It determines a high degree of similarity between two users, regardless of their rating differences. (2) It doesn't utilize all ratings provided by the users. (3) It cannot find the relationship between users if the number of common items is not enough. Adjusted Cosine (ACOS) (Wang et al. 2017) was proposed to overcome these drawbacks, but it also fails in calculating the effective similarity when users' cardinality is small. Another traditional similarity measure is Pearson Correlation Coefficient (PCC) (Senior 2017) which determined the similarity by considering only the co-rated items. Still, its efficiency is impaired when the number of co-rated items becomes less. Like the COS measure, PCC does not consider the global preference of the users. Some variants of PCC (Al-bashiri et al. 2017), such as CPCC, and SPCC,

Table 1 Existing Similarity Measures

Measures	Mathematical expression	References
COS	$\frac{\sum_{i=1}^{i'}\left(r_{u_{a,i}}\right)\left(r_{u_{b,i}}\right)}{\sqrt{\sum_{i=1}^{n}(r_{u_{a,i}})}\sqrt{\sum_{i=1}^{n}(r_{u_{b,i}})}}$	Ma et al. (2007)
PCC	$\frac{\sum_{i=1}^{i'}\left(r_{u_{a,i}}-r_{\overline{u_a}}\right)\left(r_{u_{b,i}}-r_{\overline{u_b}}\right)}{\sqrt{\sum_{i=1}^{i'}\left(r_{u_{a,i}}-r_{\overline{u_a}}\right)^2}\cdot\sqrt{\sum_{i=1}^{i'}\left(r_{u_{b,i}}-r_{\overline{u_b}}\right)^2}}$	Ahn (2008)
Jaccard	$\frac{\lvert I_{u_a}\rvert\cap\lvert I_{u_b}\rvert}{\lvert I_{u_a}\rvert\cup\lvert I_{u_b}\rvert}$	Liu et al. (2014)
MSD	$1-\frac{\sum_{i=1}^{i'}\left(r_{u_{a,i}}-r_{u_{b,i}}\right)^2}{\lvert i'\rvert}$	Liu et al. (2014)
JMSD	$\mathbf{sim}\left(\mathbf{u_a},\mathbf{u_b}\right)^{\mathbf{Jaccard}}*\mathbf{sim}(\mathbf{u_a},\mathbf{u_b})^{\mathbf{MSD}}$	Liu et al. (2014)
NHSM	$\mathbf{sim}(\mathbf{u_a},\mathbf{u_b})^{\mathbf{JPSS}}*\mathbf{sim}(\mathbf{u_a},\mathbf{u_b})^{\mathbf{URP}}$	Wang et al. (2017)
TMJ	$\mathbf{sim}(\mathbf{u_a},\mathbf{u_b})^{\mathbf{Triangle}}*\mathbf{sim}(\mathbf{u_a},\mathbf{u_b})^{\mathbf{Jaccard}}$	Sun et al. (2017)
RJaccard	$\begin{cases}\frac{1}{1+\frac{1}{\lvert I_{u_a}\rvert\cap\lvert I_{u_b}\rvert}+\frac{\overline{\lvert I_{u_a}\rvert}}{1+\overline{\overline{\lvert I_{u_a}\rvert}}}+\frac{1}{1+\overline{\overline{\lvert I_{u_b}\rvert}}}}, & \text{if } \left\lVert I_{u_a}\rvert\cap\lvert I_{u_b}\right\rVert\neq 0\\ 0, & \text{Otherwise}\end{cases}$	Bag et al. (2019)
RJMSD	$\mathbf{sim}(\mathbf{u_a},\mathbf{u_b})^{\mathbf{RJaccard}}*\mathbf{sim}(\mathbf{u_a},\mathbf{u_b})^{\mathbf{MSD}}$	Bag et al. (2019)
IPWR_Variance	$\alpha * \text{RPB\_Var}\left(u_a,u_b\right)+\beta * \text{Sim IPCC}$	Ayub et al. (2019)
IPWR_SD	$\alpha * \text{RPB\_SD}\left(u_a,u_b\right)+\beta * \text{Sim IPCC}$	Ayub et al. (2019)
Rating_Jaccard	$\begin{cases}\frac{N_T(u_a,u_b)+1}{\lvert I_{u_a}\rvert\cap\lvert I_{u_b}\rvert}, & \text{if } r_{u_{a,i}}=r_{u_{b,i}}\\ N_T\left(u_a,u_b\right), & \text{Otherwise}\end{cases}$	Ayub et al. (2020)
Rating_Jaccard_RPB	$\text{sim}\left(u_a,u_b\right)^{\text{Rating\_Jaccard}} * \text{RPB}\left(u_a,u_b\right)$	Ayub et al. (2020)

have been suggested to overcome these drawbacks. All the PCC and COS variants suffered either from the cold-start, sparsity, or both. The Jaccard coefficient (Sun et al. 2017) is the ratio of common ratings to all existing ratings. This technique only considers the common ratings. Unlike this, Mean Squared Difference (MSD) (Sun et al. 2017) measure considers only absolute ratings and ignores the proportion of common ratings. In Jaccard-Mean Squared Difference (JMSD) measure (Wang et al. 2017), Jaccard coefficient is combined with the MSD measure. It suffers from the local information and utilization of rating problems. Based on the Jaccard, PSS, and URP coefficients, a New Heuristic Similarity Measure (NHSM) is presented (Al-bashiri et al. 2017). It improvesthe system performance by eliminating the possibility of low similarity calculations despite having the same rating between users. This measure also fails (1) When more sparse entries are presents in the dataset (2) It does not utilize all users' ratings. (3) Similarity computation formula used in NHSM is complex.

Among the traditional similarity measures, the Jaccard is one of the popular and most frequently used similarity measure as it improves the system performance and give weightage to the common ratings. Several researchers recently used this measure to generate a new similarity measure. For instance, Sun et al. (2017) integrating the Jaccard and triangle similarity and proposed a Triangle Multiplying Jaccard (TMJ) similarity measure. It considers both the length and angle of the rating vectors between users. Still, it fails because, it does not consider the user's global preference. Based on the Jaccard measure, Bag et al. (2019) proposed two similarity measures: Relevant Jaccard (RJaccard) and Relevant Jaccard Mean Squared Difference (RJMSD). Their drawbacks are: (1) They compute inaccurate similarity when both the users rated the items with equal ratings. (2) They only consider the frequency of co-rated and non-co-rated items and ignore the similarity computation intensity. Apart from this, Ayub et al. (2019) also proposed the IPWR_Variance and IPWR_SD. The author integrated improved PCC with rating preference behavior (RPB) to calculate the similarity. Besides this, Ayub et al. (2020) also proposed two effective models, Rating-Jaccard and Rating-Jaccard-RPB. The new similarity models computed inaccurate similarity when the users did not have equal rating items. To overcome the sparsity issue of the CF technique, a new method improvised Gower's Jaccard (iGJ) is proposed in this paper. In the next section, we focussed on the time decay function used in CF technique.

2.2 Time decay functions in collaborative filtering-based recommendation

Classic recommendation methods utilize the rating information to calculate the similarity whereas, time information is not considered. Therefore, the traditional recommendation algorithms may not generate the appropriate nearest neighbor set for the target user. In this case, the recommendation outcomes may have low precision. To overcome this, an time-weighted recommendation system is presented. Since the user's interests change over time, the same item may receive different ratings at various periods. Therefore, several researchers used the time function in their CF-based methods. In the implication of the time decay function, two things are crucial: (1) Selection of the appropriate time decay function and (2) The level at which the time decay functions are implemented. This section discusses the previous study on the time decay functions and the level at which they can be applied. The list of popular time decay functions is given in Table 2.

The first time-based recommendation algorithm was developed by Zimdars et al. (2013), who reframed the recommendation issue as a time series prediction problem. Nowadays, most of the subsequent research is cantered on time-based recommendations. Most time decay functions include Exponential, Power, Logistic, Convex, Concave and Linear (Ding and Li 2005; Larrain et al. 2015; Xu et al. 2019).

Since the taste of users changes over time and old data becomes obsolete, the relevance of time cannot be ignored in the accuracy of prediction algorithms (Ding and Li 2005). Lee et al. (2008) developed a pseudo-rating CF approach based on implicit feedback data. The author considered the user's purchase time and the item's rating time for finding the weight decay to improve suggestion accuracy. Gong and Cheng (2008) implemented a technique for analysing the user's interest change with the CF model. In this, a predetermined weight is used to decay all users' ratings based on item rating time. Xia et al. (2010) proposed a dynamic item-based recommendation system using concave, convex, and linear time decay functions. Zheng and Li (2011) used a power decay function to improve the performance of a tag-based recommender after filtering their data based on the recency of tagging interactions. Wu et al. (2012) integrated user and item-based collaborative filtering with the power decay function for social tagging label prediction in a digital

Table 2 Existing Time Decay Functions

Time function	Mathematical expression	References
Exponential	$e^{-\mu\lvert \mathrm{T}_{\mathrm{u_a,k}}-T_{u_b,k}\rvert}$	Ding and Li (2005)
Concave	$\alpha^{\lvert T_{u_a,k}-T_{u_b,k}\rvert}$	Xia et al. (2010)
Convex	$1-\beta^{t-\lvert T_{u_a,k}-T_{u_b,k}\rvert}$	Xia et al. (2010)
Linear	$1-\frac{\lvert T_{u_a,k}-T_{u_b,k}\rvert}{t}\cdot\gamma$	Xia et al. (2010)
Power	$\lvert T_{u_a,k}-T_{u_b,k}\rvert^{-\omega}$	Larrain et al. (2015)
Logistic	$\frac{1}{1+e^{-\lambda*\lvert T_{u_a,k}-T_{u_b,k}\rvert}}$	Zhang et al. (2019)

library. Li et al. (2013) considered the time component and proposed a time weight iteration model based on the principle of memory. Huang and Song (2014) enhanced a tag-based recommender by using a two-step filtering method that used a linear decay function to simulate the recency effect of interactions. Chen et al. (2021) expand the concept of human brain memory to describe the degree of a user's interests (i.e., immediate, short-term, or long-term) and present the Dynamic Decay Collaborative Filtering (DDCF) method to modify the decay function depending on users' actions.

These temporal decay functions can be used at three separate stages of the recommendation process: Similarity Computation (SC), Rating Matrix (RM), and Prediction (P). For instance, Ma et al. (2016) applied exponential function at the prediction level to predict the time-weighted ratings. In this, author uses a hierarchical structure between items to improve similarity. Xu et al. (2019) applied the exponential function with improved ACOS functions at the similarity computation level. These current time-dependent recommendation algorithms generally add time factors in the training phase. Apart from these, a time weighting similar user selection technique is presented in Zhang et al. (2019) that employs the logistic function to weight the scores of users and items. In this, initially the evaluation time of the historical score is recorded, and then the logistic function is adopted to calculate the time weighting coefficient according to the time. In addition, some researchers use time-relative models to improve the quality of recommendations in the recommendation process. For instance, an opportunity model to estimate the probability of purchasing a product at a specific time was proposed (Wang and Zhang 2013).

To the best of our knowledge, only a single decay function is used in most of the literature work, to evaluate the changes in user preference. While a single decay function may not be sufficient to reflect the users' preference changes, we studied multiple decay functions in this paper. In addition, in the literature , the implementation of the time decay function only at one level (of the recommendation process) is described. In this paper, we applied multiple time decay functions at several levels of the recommendation process, i.e., similarity computation, rating matrix, and prediction levels and find out experimentally the level at which the results are more accurate. Table 3 lists all the notations used in the paper.

Table 3 Symbol Description

Notation	Description
$u_{a,}$ $u_{b,}$ $u_{t,}$ $u_{o,}$	The users u_a, u_b, u_t and u_o, where $u_a \neq u_b \neq u_k \neq u_o$,
i, k	The items i and k, where $i \neq k$
r_{u_a},r_{u_b}	The rating of user
$r_{u_{a,i}}$,$r_{u_{b,i}}$, $r_{u_{o,i}}$,$r_{u_{a,k}}$,$r_{u_{b,k}}$	The rating given by a user on an item
$R_{u_{a,i}}$	The time weighted rating
$P_{u_t,i}$	The rating predicted by a user on an item
$\overline{r_{u_a}}$, $\overline{r_{u_b}}$, $\overline{r_{u_c}}$	The average rating of the user
μ_k	The average rating of item
$\|I_{u_a}\|$, $\|I_{u_b}\|$	The cardinality of the user
$\|\bar{I}_{u_a}\|$, $\|\bar{I}_{u_b}\|$	The cardinality of the un-co-rated items of user
NBR (u_t)	The number of neighbors of a user
$N_T(u_a, u_b)$	The number of ratings that are equal in absolute value
r_{med}, r_{max},r_{min}	The median, maximum and minimum value in the rating scale
sim (u_a, u_b)	The similarity between two users
M, N, i'	The total number of users, items and co-rated items respectively
R	The total number of ratings
R_k	The difference between maximum and minimum ratings of an item
$D_{u_a u_b k}$	The distance between two users for an item
$W_{u_a u_b k}$	The weight given by two users for an item
$\|E\|$	The cardinality of the testing set
$\propto$, β, Y, μ, λ, ω	Tuning Parameters
$tu_a k$, tu_bk	Time stamp of user u_a and u_b when they rated the item k
t	Value of the largest time interval in the training dataset

3 Motivation for the new similarity measure

In the CF approach, researchers have proposed many similarity measures. This section analyses the major shortcomings of existing similarity measures, stated in Table 1.

The cosine measure computes high correlation despite having significant differences between their ratings. For instance, it computes the maximum similarity between user 1 (0,0,1,0,0,0) and user 2(0,0,5,0,0,0). In contrast, their rating preferences indicate that incorrect similarity is calculated between these users, as both have rated only one item (I3). Apart from this, the PCC measure calculates zero similarity between user 3(1,0,1,2,0,0) and user 4 (5,0,1,0,0,0), even they give similar ratings to certain items. The MSD measure ignores the proportion of common ratings. JMSD measure computes less similarity between user 5 (4,0,3,0,0,0) and user 6 (4,3,3,4,4,3) in comparison to user 4 (5,0,1,0,0,0) and user 5 (4,0,3,0,0,0). This is an inaccurate similarity calculation as users 5 and 6 have more similar rating items than users 4 and 5. The drawback of the NHSM measure is that it has a complex formula for similarity computation, and it computes zero similarity between user 7(0,1,0,0,0,0) and user 8 (0,3,0,5,5,3). Similarly, TMJ computes zero similarity between user 5(4,0,3,0,0,0) and user 6(4,3,3,4,4,3). The RJaccard and RJMSD both compute inaccurate similarity when both the users rated the items with equal ratings. They only consider the frequency of co-rated and un-co-rated items and ignore the intensity for the similarity computation. RJaccard and RJMSD calculates the same similarity between user 3(1,0,1,2,2,0)–6(4,3,3,4,4,3), and 6 (4,3,3,4,4,3)–8 (0,3,0,5,5,3), while their rating preferences indicate that similarity between users 6–8 should be more. In the same way, Rating_Jaccard and Rating_Jaccard_RPB computed inaccurate similarity when the users do not have equal rating items. For example, consider user 7(0,1,0,0,0,0) and user 8(0,3,0,5,5,3), Rating_Jaccard and Rating_Jaccard_RPB computes zero similarity in this case. Furthermore, IPWR_Variance and IPWR_SD ignore the user's global preference.

To overcome the drawback of the existing measures, a new similarity measure is proposed in the next section. Also, these measures use the historical ratings while computing similarity, but the user preferences change over time; therefore, with the help of proposed method and time function, a time-based recommendation system is also developed in this paper.

4 Proposed similarity method

The traditional CF based RS algorithms do not give importance to the fact that the user's interest changes with time. Therefore, considering time becomes crucial for the performance of the recommendation system. Taking this into consideration, the research proposes to evaluate various time decay functions by applying them at various stages of the recommendation process using a novel similarity measure iGJ.

4.1 Improvised Gower Jaccard (iGJ) similarity coefficient

In 1971, J.C. Gower's introduced the most common proximity measure for mixed data types, known as Gower's coefficient (Podani 1999). It can work on heterogeneous data such as binary, categorical, and ordinal data and can also be applied to quantitative and qualitative data. It also has the advantage of working effectively when some ratings in the data matrix are missing (Fontecha et al. 2014). Podani (1999) developed an enhanced version of Gower's coefficient, which assessed the similarity between ordinal characters. ben Ali and Massmoudi (2013) presented a study on Gower's coefficient and k-means clustering. Fontecha et al. (2014) used Gower's coefficients to create a novel mobile service system to increase the accuracy of frailty assessments in an elderly population. Gower's coefficient is used to calculate similarity in the following way:

$$\mathrm{sim}\left(\mathrm{u_a},\mathrm{u_b}\right)^{\mathrm{Gower's}}=\frac{\sum_{k=1}^{n}\mathrm{D}_{\mathrm{u_a u_b k}}\cdot\mathrm{W}_{\mathrm{u_a u_b k}}}{\sum_{k=1}^{n}\mathrm{W}_{\mathrm{u_a u_b k}}} \tag{1}$$

Equation 1 calculates the similarity value between 0 and 1, where zero indicates the lowest similarity and one corresponds to high similarity. Calculating the zero similarity is feasible only when no co-rated items exist between two users. Its value cannot be greater than one. For each user item, the values of $\mathrm{W}_{\mathrm{u_a u_b k}}$ and $\mathrm{D}_{\mathrm{u_a u_b k}}$ for various categories of data variables may be determined as follows:

For Categorical Variables:

$$\begin{cases}\mathrm{W}_{\mathrm{u_a u_b k}}=1, & \text{if both the rating value are known}\\ 0, & \text{Otherwise}\end{cases}$$

$$\begin{cases}\mathrm{D}_{\mathrm{u_a u_b k}}=1, & \text{if both the rating value are equal}\\ 0, & \text{Otherwise}\end{cases}$$

For Binary Variables:

$$\begin{cases}\mathrm{W}_{\mathrm{u_a u_b k}}=0 \text{ and } \mathrm{D}_{\mathrm{u_a u_b k}}=0, & \text{if both the rating value is zero}\\ \mathrm{W}_{\mathrm{u_a u_b k}}=1 \text{ and } \mathrm{D}_{\mathrm{u_a u_b k}}=0, & \text{if one rating value is zero}\\ \mathrm{W}_{\mathrm{u_a u_b k}}=1 \text{ and } \mathrm{D}_{\mathrm{u_a u_b k}}=1, & \text{if both the rating value is one}\end{cases}$$

For Ordinal Variables:

$$\begin{cases} \mathrm{W}_{\mathrm{u_a u_b k}} = 1, & \text{if both the rating value are known} \\ 0, & \text{Otherwise} \end{cases}$$

$$\begin{cases} \mathrm{D}_{\mathrm{u_a u_b k}} = 1 - \dfrac{\left|\mathrm{r}_{\mathrm{u_{a,k}}} - \mathrm{r}_{\mathrm{u_{b,k}}}\right|}{\mathrm{R_k}} \end{cases} \tag{2}$$

Gower's coefficient was applied to the user-item rating matrix to calculate similarity among users, which helps determine the neighbours and generate the prediction. It is observed that Gower computes incorrect similarity in some instances during the similarity calculation phase. This is the case when the numerator and denominator become equal in Eq. 2. For example, consider two users' user-1 (5,3) and user-2 (2,1); Gower's coefficient calculates 0 similarities among these users, which is false. To address this, we suggest an improvised Gower (iG) coefficient in Jain et al. (2021). The following formula is used to compute the similarity using the improvised Gower's coefficient:

$$\mathrm{Sim}\left(\mathrm{u_a}, \mathrm{u_b}\right)^{\mathrm{Improvised\_Gower's}} = \frac{\sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{D}_{\mathrm{u_a u_b k}}^{\mathrm{Improvised}} \cdot \mathrm{W}_{\mathrm{u_a u_b k}}}{\sum_{\mathrm{k=1}}^{\mathrm{n}} \mathrm{W}_{\mathrm{u_a u_b k}}} \tag{3}$$

where,

$$D_{u_a u_b k}^{improvised} = \begin{cases} 1 - \dfrac{\left|r_{u_{a,k}} - r_{u_{b,k}}\right| - random(0, 0.5)}{R_k}, & \text{If } r_{u_{a,k}} - r_{u_{b,k}} = R_k \\ 1 - \dfrac{\left|r_{u_{a,k}} - r_{u_{b,k}}\right|}{R_k}, & \text{Otherwise} \end{cases}$$

The improvised Gower's coefficient computes the non-zero similarity between users who have rated at least one common item.

To give weightage to the common items, the Jaccard measure is combined with the improvised Gower's coefficient. The Jaccard coefficient is defined as the ratio of the intersection to the union of sample sets. It considers only the number of common ratings among two users while computing similarity. Similarity using the Jaccard coefficient can be investigated as follows:

$$\mathrm{sim}\left(\mathrm{u_a}, \mathrm{u_b}\right)^{\mathrm{Jaccard}} = \frac{\left|\mathrm{I}_{\mathrm{u_a}}\right| \cap \left|\mathrm{I}_{\mathrm{u_b}}\right|}{\left|\mathrm{I}_{\mathrm{u_a}}\right| \cup \left|\mathrm{I}_{\mathrm{u_b}}\right|} \tag{4}$$

Thus, the proposed similarity measure improvised Gower-Jaccard (iGJ) is the combination of improvised Gower's and Jaccard's similarity measures. The main reason for selecting Gower's coefficient is that it works well even when some entries are missing in the data matrix. Moreover, most of the ratings are absent in the user-item rating matrix dataset, requiring the incorporation of all the common ratings for similarity computation. For this, Jaccard is the most suitable measure. The similarity computation formula of the proposed improvised Gower's Jaccard (iGJ) similarity measure is rendered in Eq. 5.

$$\mathrm{sim}\left(\mathrm{u_a}, \mathrm{u_b}\right)^{\mathrm{iGJ}} = \mathrm{sim}\left(\mathrm{u_a}, \mathrm{u_b}\right)^{\mathrm{Improvised\ Gowers}} * \mathrm{sim}\left(\mathrm{u_a}, \mathrm{u_b}\right)^{\mathrm{Jaccard}} \tag{5}$$

The proposed measure iGJ considers all available rating data while calculating similarity, thereby reducing the sparsity problem.

4.2 Theoretical performance evaluation of iGJ measure

This section states that the proposed iGJ successfully overcomes the existing methods' shortcomings (described in Sect. 3).

As discussed in Sect. 3, the cosine measure computes the maximum similarity between users when they have only one co-rated item. iGJ overcomes this flaw and computes non-zero similarity (0.125) in this case. Additionally, the proposed iGJ method calculates the non-zero similarity between users and removes the lack of zero similarity computation by some measures discussed in Sect. 3. For example, iGJ calculates the non-zero similarity between users 3 and 4 (=0.125), between users 5 and 6 (=0.333), and between users 7 and 8 (=0.062). iGJ also overcames the drawback of MSD measure by considering the common and absolute ratings. It also considers the user's global preference. It computes better similarity between users 5 and 6 (0.333) as compared to users 4 and 5 (=0.219) and overcomes the flaw of the JMSD measure. It also overcomes the weakness of RJaccard and RJMSD measures by computing more similarity between users 6 and 8(0.333) than users 3 and 6(0.194). The Rating_Jaccard and Rating_Jaccard_RPB computed inaccurate similarity when the users do not have equal rating items. iGJ computed 0.062 similarity in this case. Thus, with the help of this discussion, we can state that our proposed method performs better than existing approaches. Section 5, evaluated the performance of iGJ on a real-world dataset.

In e-commerce, it is found that time is also an important factor because the choice of individuals changes more frequently with time. So, to improve system efficiency and give importance to the recent data, six-time decay functions with our proposed measure in the collaborative filtering technique is used. These time decay functions can be applied in three ways, and a discussion is presented in the next section.

4.3 iGJ coefficient with time decay functions

In CF, a time decay function can be applied at the following three places of the recommendation process: (1) Construction of Rating Matrix, (2) Similarity Computation (3)

Prediction of rating. These three steps are very important in CF technique to make the recommendation, therefore various time decay functions as listed in Table 2 are applied at these three levels to determine their accuracy. In each TDF, different tuning parameters are used. The value of those parameters can be obtained through various experiments. Since similarity computation is crucial in the recommendation process, we first applied all the time decay functions at the similarity computation level to obtain the best values of the tuning parameters. In the next two levels, i.e., rating matrix and prediction levels, we used the best values of the tuning parameters of the time function. A detailed definition of each of them is given below:

4.3.1 *Time decay function at similarity computation level (TDF@SCL)*

In this approach, iGJ with various time decay functions are applied at the similarity computation level of the recommendation system. To obtain similarity, the actual ratings present in the dataset are used and then iGJ similarity measure with various TDF given in Table 2 are applied. The appropriate similarity is calculated as follows:

$$\mathrm{sim}\left(\mathrm{u_a},\mathrm{u_b}\right)^{\mathrm{iGJ\_TDF}} = \mathrm{sim}\left(\mathrm{u_a},\mathrm{u_b}\right)^{\mathrm{iGJ}} * \mathrm{TDF} \tag{6}$$

where TDF (Δt) is a time decay function, it can be exponential, power, logistic, linear, concave, and convex. After calculating the similarity using Eq. 6, the nearest neighbors of the target users are determined to generate the prediction using the formula given below.

$$P_{u_t,i} = \overline{r_{u_t}} + \frac{\sum_{u_o \epsilon NBR(u_t)} SIM\left(u_t,u_o\right) * \left(r_{u_o,i} - \overline{r_{u_o}}\right)}{\sum_{u_o \epsilon NBR(u_t)} SIM\left(u_t,u_o\right)} \tag{7}$$

4.3.2 *Time decay function at rating matrix level (TDF@RML)*

In this approach, we first convert actual ratings of the dataset into time-weighted ratings using the TDF and then apply the iGJ similarity measure to them to obtain the similarity.

$$R_{u_a,i} = r_{u_a,i} * \mathrm{TDF} \tag{8}$$

In the time-based RS, the ratings that are closer to the current time achieved the larger weighting coefficient. After calculating the similarity and determining the neighbors, a prediction is made for each target user, using Eq. 7.

4.3.3 *Time decay function at prediction level (TDF@PL)*

The prediction step is performed in the CF-based RS after calculating the similarity and determining the neighbors. In this approach, we first calculate the similarity using our proposed method (Eq. 5) and then determine each target user's neighbor set. After obtaining the neighbors, we incorporated the time decay function in the prediction formula to get the predicted ratings. The prediction formula (Jain et al. 2020) for this approach is given in Eq. 9.

$$P_{u_t,i} = \overline{r_{u_t}} + \frac{\sum_{u_o \epsilon NBR(u_t)} SIM\left(u_t,u_o\right) * \left(r_{u_o,i} - \overline{r_{u_o}}\right) * \mathrm{TDF}}{\sum_{u_o \epsilon NBR(u_t)} SIM\left(u_t,u_o\right)} \tag{9}$$

In the next section, the efficiency of iGJ is displayed with a comparison of the traditional and recently proposed method. In addition, the performance of each time decay function is also examined at various levels of the recommendation process. Figure 1 shows the experimental model of the traditional and time-based collaborative filtering method and the algorithm for the same is depicted below.

Algorithm 1: Rating Prediction by integrating iGJ and time decay functions

```
Input: User – Item Rating Matrix
1: $sim(u_a,u_b)^{Proposed\ measure}$ = [], $P_{u_t,i}$ = []
2:   for the user $u_a$ = 1 to M do
3:     for the user $u_b$ = 1 to M do
4:         for the item k = 1 to N do
5:             if $r_{u_{a,k}} - r_{u_{b,k}}$ != 0 then
6:                 if $r_{u_{a,k}} - r_{u_{b,k}} = R_k$ then
7:                     $D_{u_a u_b k}^{Imporvised} = 1 - \frac{|r_{u_{a,k}} - r_{u_{b,k}}|}{1+R_k}$
8:                     $W_{u_a u_b k} = 1$
9:                 else
10:                    $D_{u_a u_b k}^{improvised} = 1 - \frac{|r_{u_{a,k}} - r_{u_{b,k}}|}{R_k}$
11:                    $W_{u_a u_b k} = 1$
12:                end if
13:        $sim(u_a,u_b)^{imrovised\_Gower's} = \frac{\sum_{k=1}^{n} D_{u_a u_b k}^{improvised} . W_{u_a u_b k}}{\sum_{k=1}^{n} W_{u_a u_b k}}$
14.        end for
15.      end for
16.  end for
17.  for the user $u_a$ = 1 to M do
18:     for the user $u_b$ = 1 to M do
19.         $sim(u_a,u_b)^{Jaccard} = \frac{|I_{u_a}| \cap |I_{u_b}|}{|I_{u_a}| \cup |I_{u_b}|}$
20.     end for
21.  end for
22. #TDF at similarity computation level
23.  for the user $u_a$ = 1 to M do
24.    for the user $u_b$ = 1 to M do
25.        $sim(u_a,u_b)^{Proposed\ measure} = sim(u_a,u_b)^{Improvised\_Gower's} \times sim(u_a,u_b)^{Jaccard} \times$ TDF
26.    end for
27.  end for
28.  for the user $u_t$ = 1 to M do
29.     for the item i = 1 to N do
30.         if $sim(u_t,u_o)$> 0 then
31.             $P_{u_t,i} = \overline{r_{u_t}} + \frac{\sum_{u_o \in NBR(u_t)} SIM(u_t,u_o) * (r_{u_o,i} - \overline{r_{u_o}})}{\sum_{u_o \in NBR(u_t)} SIM(u_t,u_o)}$
32.         else
33.             $P_{u_t,i} = 0$
34.         end if
35.     end for
36.  end for
37. #TDF at rating matrix level
38.  $R_{u_a,i} = r_{u_a,i}$ * TDF
39.  for the user $u_a$ = 1 to M do
40.    for the user $u_b$ = 1 to M do
41.        $sim(u_a,u_b)^{Proposed\ measure} = sim(u_a,u_b)^{Improvised\_Gower's} \times sim(u_a,u_b)^{Jaccard}$
42.    end for
43.  end for
44.  for the user $u_t$ = 1 to M do
45.     for the item i = 1 to N do
46.         if $sim(u_t,u_o)$> 0 then
47.             $P_{u_t,i} = \overline{r_{u_t}} + \frac{\sum_{u_o \in NBR(u_t)} SIM(u_t,u_o) * (r_{u_o,i} - \overline{r_{u_o}})}{\sum_{u_o \in NBR(u_t)} SIM(u_t,u_o)}$
48.         else
49.             $P_{u_t,i} = 0$
50.         end if
51.     end for
52.  end for
53. #TDF at Prediction level
54.  for the user $u_a$ = 1 to M do
55.    for the user $u_b$ = 1 to M do
56.        $sim(u_a,u_b)^{Proposed\ measure} = sim(u_a,u_b)^{Improvised\_Gower's} \times sim(u_a,u_b)^{Jaccard}$
57.    end for
58.  end for
59.  for the user $u_t$ = 1 to M do
60.     for the item i = 1 to N do
61.         if $sim(u_t,u_o)$> 0 then
62.             $P_{u_t,i} = \overline{r_{u_t}} + \frac{\sum_{u_o \in NBR(u_t)} SIM(u_t,u_o) * (r_{u_o,i} - \overline{r_{u_o}}) * TDF}{\sum_{u_o \in NBR(u_t)} SIM(u_t,u_o)}$
63.         else
64.             $P_{u_t,i} = 0$
65.         end if
66.     end for
67.  end for
Output: Value of predicted ratings
```

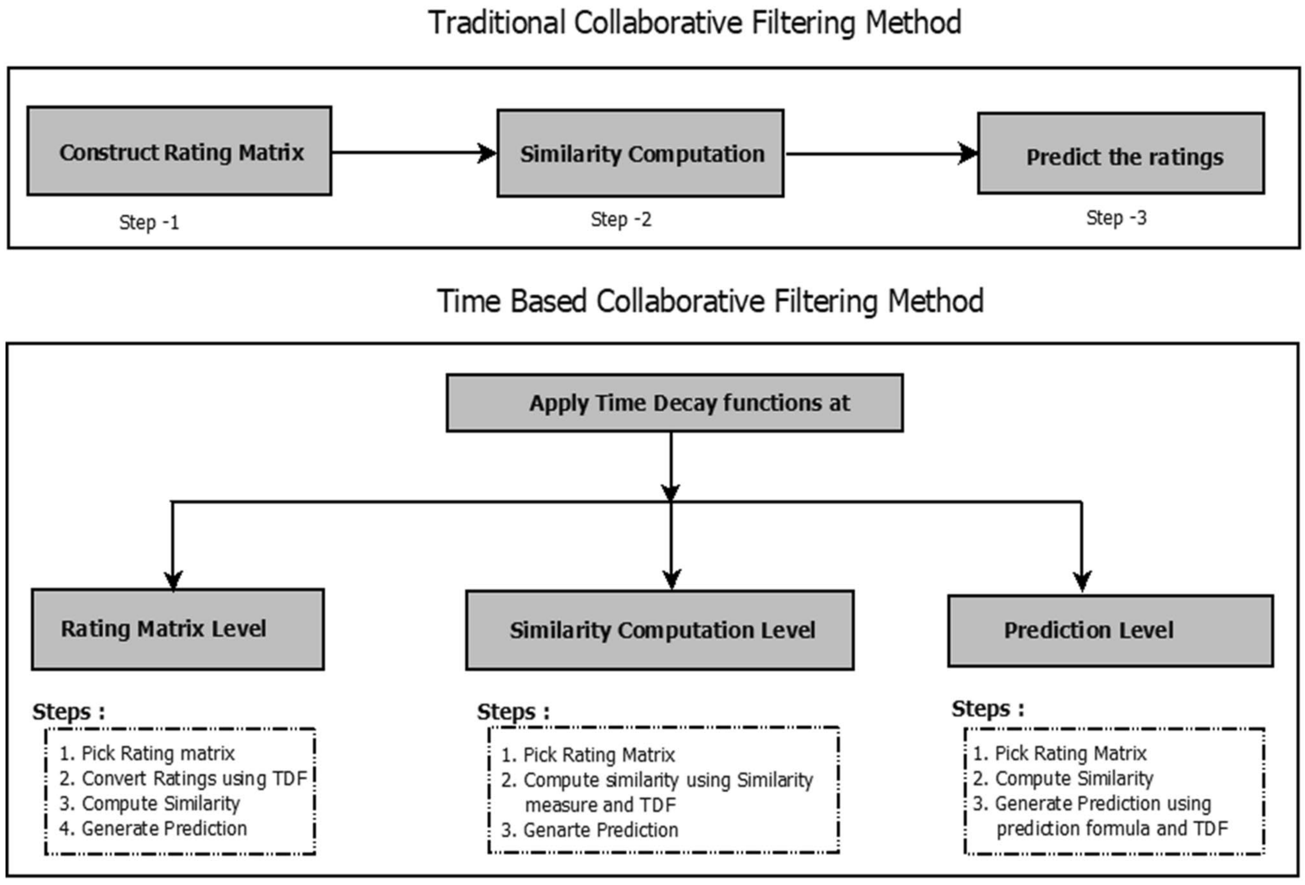

Fig. 1 Traditional versus time based collaborative filtering method

5 Experimental results

The proposed algorithm is implemented with the existing algorithms on the Scientific Python Programming Framework Spider 3.3.3. The systems architecture is Intel(R) Xeon(R) CPU E5-2680 v2@2.80 GHz, 64 GB RAM, and Microsoft Windows 10 Education operating system. Initially, the experimental setup includes a description of the dataset used, followed by the prediction algorithm. In the following subsection, evaluation metrics are described. The proposed method's efficiency is examined in the last segment.

5.1 Datasets

MovieLens-100K, Epinions, and Amazon Magazine datasets are used for performance evaluation of iGJ. Table 4 presents the details of datasets.

MovieLens-100k (Tan and He 2017): The Group Lens Research Group created the MovieLens-100k dataset at the University of Minnesota. In the dataset, every user rated at least 20 movies on a scale of 1–5, with one being worst and five being best. This dataset covers a total of 8-months of data.

Table 4 Dataset Description

	Movie lens-100k	Epinions	Amazon magazine subscription
Users (M)	943	40,163	2428
Items (N)	1682	139,738	72,098
Ratings (R)	100,000	664,823	89,689
Sparsity index (%)	93.70	99.99	99.95
Time range	Sep 1997–April 1998	July 1999–May 2011	May 1996–Oct 2018

Epinions (Manochandar and Punniyamoorthy 2020): The sparsity level of this dataset is very high. It covers almost 12 years of data. The dataset ratings are present on a scale of 1–5. It's a system for rating goods and services from various sources.

Amazon Magazine Subscription (Ni et al. 2019): The ratings in the dataset are in the range of 1–5, where one being worst and five being best. This dataset covers more than 22 years of data.

5.2 Evaluation metrics

Determining the accuracy of an algorithm in a recommendation system is a challenging issue because algorithms behave differently depending on the input dataset. An evaluation metric (Nguyen et al. 2021) is used to test such algorithms, which can be classified mainly into Predictive accuracy metrics (MAE, RMSE) and Classification accuracy metrics (Precision, recall, F1-measure, and accuracy). Predictive accuracy metrics is applied to test the algorithms. In MAE and RMSE, the predicted ratings are compared with the actual ratings, and their lower values indicate that method's better performance. The formulas for the MAE and RMSE metrics are given below:

$$\mathrm{MAE} = \frac{1}{|E|}\sum_{i=1}^{|E|}\left|P_{u_{t,i}} - r_i\right| \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{|E|}\sum_{i=1}^{|E|}\left|P_{u_{t,i}} - r_i\right|} \tag{11}$$

5.3 Experimental results

In this section, the experiment results are displayed in two parts. In the first part, the comparison between the proposed similarity measure iGJ and several other existing measures is shown. In the next part, the results of applying the various time decay functions with iGJ at various levels of the recommendation process, i.e., rating matrix, similarity computation levels, and the prediction level, are shown. Also, the number of neighbors is an important parameter to evaluate the performance of CF-based RS; therefore, in both the experiments, we assess the results at different-2 neighbors, i.e., 20, 60, 100, 150, 200. For the evaluation, the MAE and RMSE metrics are utilized.

5.3.1 Performance evaluation of iGJ

This experiment aims to test the predictive accuracy of the iGJ with the existing similarity measures, i.e., PCC, Cosine, Jaccard, NHSM Rating_Jaccard, RJaccard, etc. For this, a very popular and most commonly used ML-100k dataset is used. The MAE and RMSE results on the are displayed in Figs. 2 and 3, respectively.

From both the figures, it is clear that the proposed method iGJ calculated lower MAE and RMSE values compared to all other methods. After the iGJ, the performance of NHSM and Jaccard measures is good. At k = 20, the IPWR_Variance, IPWR_SD, Rating_Jaccard, and Rating_Jaccard_RPB are almost similar in MAE comparison, while RJMSD shows poor results for the remaining k's value. In RMSE comparison, at k = 20, Rating_Jaccard_RPB gives the worst results, while the RJSMD measure gives the worst results for the remaining values of k's.

Through this experiment, we can state that the iGJ outperforms the other methods as its calculated MAEs and RMSEs values are lower than other methods for every k's values. In the next section, we display the results of applying the several time decay functions with iGJ on various level of recommendation process.

5.3.2 Experimental results of applying iGJ with time decay function (iGJ-TDF)

In the CF technique, three steps are very important to make the recommendation; construction of the user-item rating matrix, similarity computation, and predicting the rating for the target users. The time functions listed in Table 2 are applied on these three levels. Since, the performance of recommendation systems largely depends upon the effective similarity calculation, henceforth, we firstly apply all the time decay functions at the similarity computation level and then apply at rating matrix and prediction levels. To test the performance, we used three datasets i.e., ML100k, Epinions, and Amazon Magazine dataset. The best results in each Table are boldfaced.

5.3.2.1 Time decay function at similarity computation level (TDF@SC) In this, iGJ measure is applied with exponential, concave, convex, linear, logistic, and power time functions at similarity computation level. A tuning parameter is used in each time decay function, whose best value in the range of 0.5 to 0.9 is obtained by experiments. Experimental results on ML-100k, Epinions, and Amazon Magazine datasets are shown in Tables 5, 6, and 7, respectively.

From the results, it is evident concave, convex, exponential, linear, logistic, and power function gives the best results at 0.9,0.5,0.5,0.7,0.9,0.6, respectively for all the three datasets. The experimental results also indicate that, among all time functions, the convex function gives better results, when applied that at the similarity computation level. Next, we applied a time function on the rating and prediction level using these best tuning parameter values, displayed through Table 8.

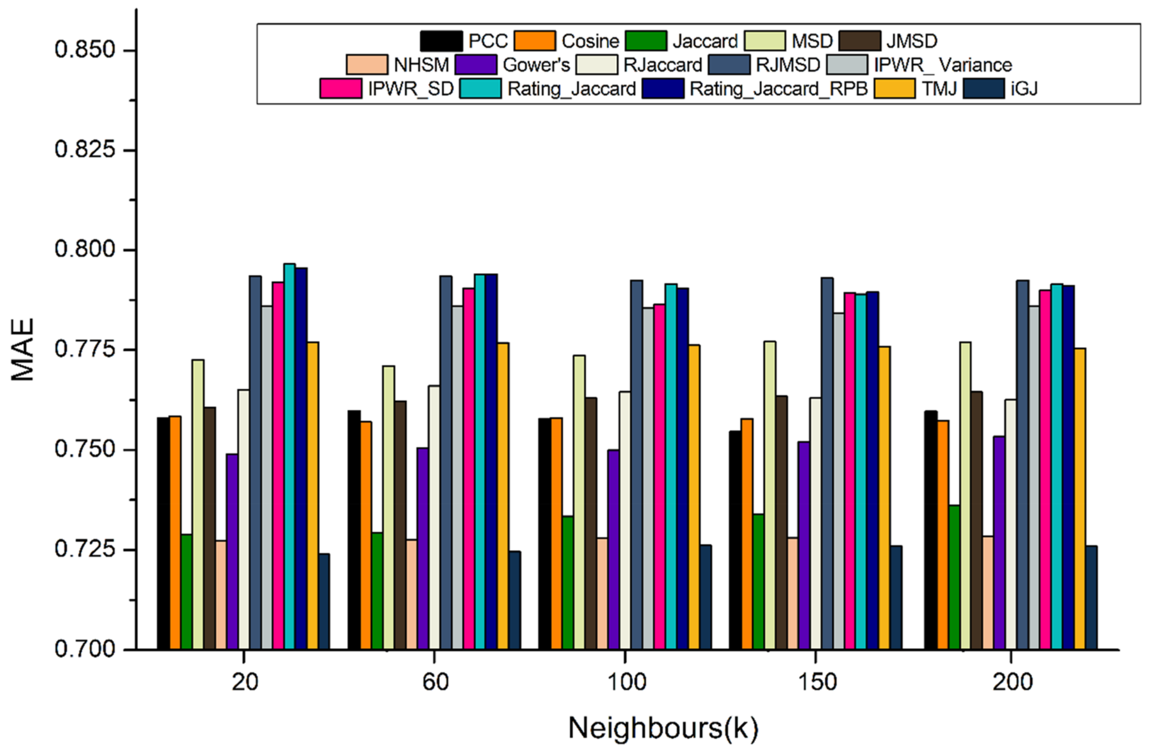

Fig. 2 MAE comparison of the proposed measure and existing methods

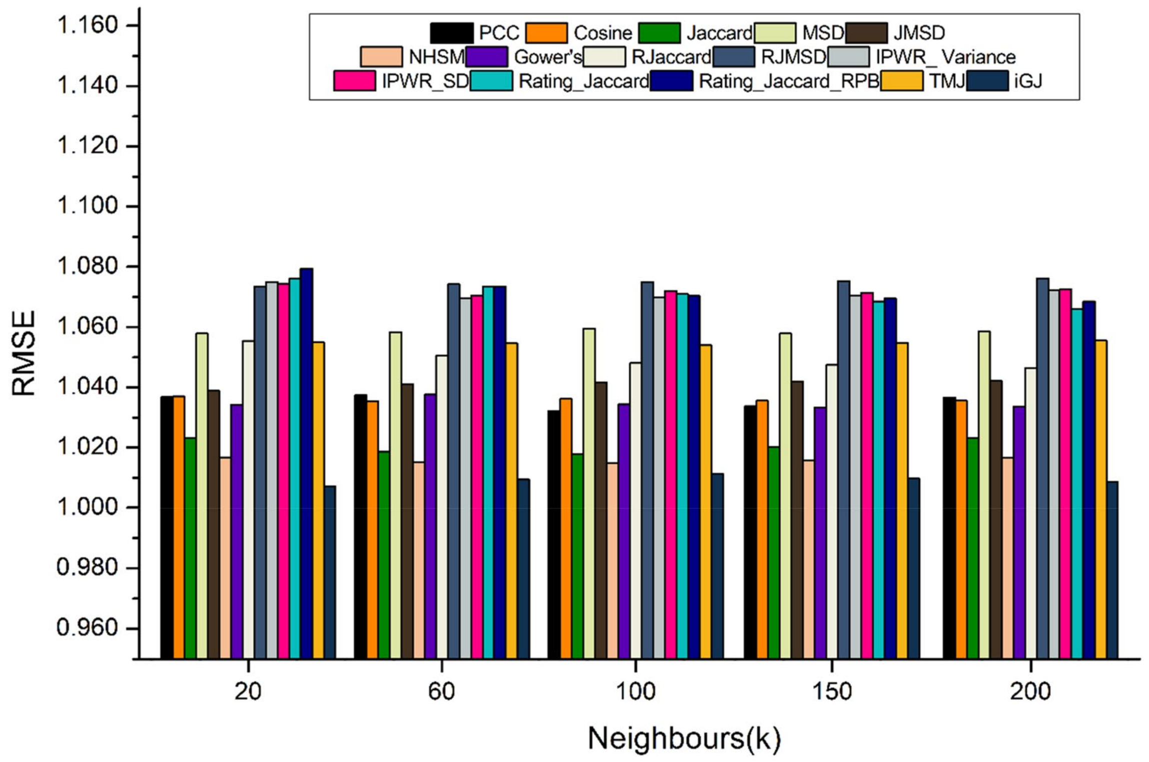

Fig. 3 RMSE comparison of the proposed measure and existing methods

Table 5 Results of applying TDF at SC level on ML-100k dataset

Dataset	Parameters	MAE					RMSE				
		20	60	100	150	200	20	60	100	150	200
Concave	α = 0.5	0.75800	0.75850	0.76050	0.76300	0.76450	1.04200	1.04150	1.04350	1.04500	1.04650
	α = 0.6	0.75700	0.75650	0.75950	0.76150	0.76350	1.04100	1.03950	1.04300	1.04400	1.04550
	α = 0.7	0.75600	0.75500	0.75800	0.76050	0.76150	1.04050	1.03750	1.04100	1.04250	1.04400
	α = 0.8	0.75300	0.75250	0.75500	0.75800	0.75950	1.03700	1.03500	1.03800	1.04050	1.04250
	α = 0.9	**0.75000**	**0.75100**	**0.75250**	**0.75600**	**0.75750**	**1.03400**	**1.03350**	**1.03550**	**1.03950**	**1.04050**
Convex	β = 0.5	**0.74800**	**0.74900**	**0.75200**	**0.75450**	**0.75800**	**1.03150**	**1.03150**	**1.03400**	**1.03750**	**1.04000**
	β = 0.6	0.74835	0.74950	0.75240	0.75500	0.75860	1.03175	1.03250	1.03450	1.03850	1.04100
	β = 0.7	0.74840	0.75000	0.75250	0.75600	0.75850	1.03180	1.03350	1.03500	1.03950	1.04050
	β = 0.8	0.75430	0.75050	0.75255	0.75650	0.75900	1.03350	1.03400	1.03550	1.03900	1.04150
	β = 0.9	0.75300	0.75100	0.75250	0.75750	0.75950	1.03700	1.03400	1.03800	1.04000	1.04200
Exponential	μ = 0.5	**0.75645**	**0.75650**	**0.75950**	**0.76150**	**0.76300**	**1.04050**	**1.03950**	**1.04200**	**1.04350**	**1.04550**
	μ = 0.6	0.75750	0.75800	0.76000	0.76250	0.76350	1.04150	1.04050	1.04250	1.04500	1.04600
	μ = 0.7	0.75800	0.75900	0.76050	0.76300	0.76450	1.04200	1.04150	1.04300	1.04500	1.04650
	μ = 0.8	0.75850	0.76100	0.76100	0.76350	0.76450	1.04250	1.04350	1.04250	1.04600	1.04700
	μ = 0.9	0.75800	0.76000	0.76100	0.76400	0.76550	1.04200	1.04350	1.04400	1.04650	1.04750
Linear	Υ = 0.5	0.75100	0.75150	0.75250	0.75600	0.75700	1.03250	1.03300	1.03500	1.03950	1.04100
	Υ = 0.6	0.75100	0.75250	0.75250	0.75600	0.75850	1.03300	1.03300	1.03450	1.03950	1.04100
	Υ = 0.7	**0.75050**	**0.75100**	**0.75150**	**0.75550**	**0.75800**	**1.03200**	**1.03250**	**1.03400**	**1.03900**	**1.04050**
	Υ = 0.8	0.75100	0.75150	0.75300	0.75700	0.75850	1.03300	1.03450	1.03650	1.03950	1.04200
	Υ = 0.9	0.75100	0.75150	0.75400	0.75750	0.75900	1.03500	1.03450	1.03700	1.04000	1.04150
Logistic	λ = 0.5	0.75028	0.75315	0.75440	0.75603	0.75803	1.03382	1.03641	1.03793	1.03924	1.04104
	λ = 0.6	0.74978	0.75288	0.75465	0.75603	0.75800	1.03370	1.03661	1.03807	1.03928	1.04108
	λ = 0.7	0.74980	0.75313	0.75465	0.75620	0.75798	1.03325	1.03659	1.03822	1.03930	1.04108
	λ = 0.8	0.74995	0.75288	0.75485	0.75595	0.75805	1.03369	1.03674	1.03825	1.03919	1.04115
	λ = 0.9	**0.74943**	**0.75208**	**0.75423**	**0.75582**	**0.75790**	**1.03295**	**1.03592**	**1.03782**	**1.03911**	**1.04100**
Power	ω = 0.5	0.75160	0.75300	0.75660	0.75848	0.76035	1.03528	1.03646	1.03952	1.04122	1.04283
	ω = 0.6	**0.75118**	**0.75275**	**0.75575**	**0.75803**	**0.75998**	**1.03490**	**1.03604**	**1.03855**	**1.04056**	**1.04268**
	ω = 0.7	0.75223	0.75373	0.75708	0.75903	0.76058	1.03575	1.03715	1.04006	1.04148	1.04304
	ω = 0.8	0.75240	0.75418	0.75750	0.75985	0.76145	1.03615	1.03751	1.04053	1.04271	1.04386
	ω = 0.9	0.75300	0.75455	0.75808	0.76068	0.76183	1.03682	1.03783	1.04080	1.04323	1.04426

The best entries are highlighted in bold

5.3.2.2 Time decay function at rating matrix level (TDF@RM) Table 8 shows the best values of the tuning parameters of each time function, which are determined through the experiment in the previous step. In this section, the results of applying time decay functions at only the best values of tuning parameters at the rating matrix level are depicted. For this, original ratings are converted into time-weighted ratings using time decay functions and then iGJ similarity measure is applied to compute the similarity among users. Experimental results given in Table 9 indicate that the performance of the power function is incomparable from all other functions on all datasets. The MAE and RMSE value of that is very low than other; hence it is superior.

5.3.2.3 Time decay function at prediction level (TDF@P) This section demonstrates the experimental results of applying time decay functions at the prediction level of the recommendation process. Like the rating matrix level, TDF is applied with the best values of the tuning parameters. The experimental results displayed in Table 10 show that the convex function gives lower MAE and RMSE than all other time functions on all three datasets.

The best values of iGJ with time functions are compared with iGJ without time function on the ML-100k, Epinions and Amazon Magazine datasets and their results are presented in Figs. 4, 5 and 6, respectively. From the results, we can conclude that at similarity computation and prediction level, convex function demonstrates best performance, while at rating matrix level power function depicts satisfactory results. With comparison to all these two, iGJ with power function at the rating matrix level gives the best results.

Table 6 Results of applying TDF at SC level on Epinions dataset

Dataset	Parameters	MAE					RMSE				
		20	60	100	150	200	20	60	100	150	200
Concave	α = 0.5	1.03889	1.04048	1.03921	1.03857	1.03810	1.43001	1.43045	1.42934	1.42901	1.42862
	α = 0.6	1.03921	1.03985	1.03842	1.03857	1.03762	1.43012	1.42990	1.42862	1.42868	1.42857
	α = 0.7	1.03889	1.04032	1.03873	1.03857	1.03810	1.42990	1.43073	1.42917	1.42923	1.42873
	α = 0.8	1.03841	1.04048	1.03842	1.03835	1.03794	1.42929	1.43079	1.42884	1.42890	1.42868
	α = 0.9	**1.03730**	**1.03953**	**1.03821**	**1.03816**	**1.03744**	**1.42812**	**1.42990**	**1.42843**	**1.42827**	**1.42812**
Convex	β = 0.5	**1.03712**	**1.03946**	**1.03837**	**1.03826**	**1.03789**	**1.42818**	**1.42821**	**1.42813**	**1.42827**	**1.42833**
	β = 0.6	1.03736	1.04080	1.03889	1.03905	1.03842	1.42856	1.43067	1.42868	1.42959	1.42862
	β = 0.7	1.03778	1.04111	1.03952	1.03905	1.03857	1.42851	1.43101	1.42934	1.42947	1.42868
	β = 0.8	1.03762	1.04096	1.03857	1.03873	1.03826	1.42823	1.43034	1.42879	1.42896	1.42857
	β = 0.9	1.03810	1.04064	1.03852	1.03859	1.03865	1.42873	1.43073	1.42884	1.42868	1.42873
Exponential	μ = 0.5	**1.03789**	**1.03916**	**1.03838**	**1.03821**	**1.03793**	**1.42939**	**1.42923**	**1.42905**	**1.42921**	**1.42901**
	μ = 0.6	1.03905	1.04000	1.03857	1.03842	1.03795	1.42973	1.42995	1.42956	1.42953	1.42945
	μ = 0.7	1.03889	1.03969	1.03873	1.03870	1.03810	1.42968	1.42973	1.42951	1.42947	1.42940
	μ = 0.8	1.03830	1.03921	1.03842	1.03839	1.03799	1.42950	1.42951	1.42951	1.42940	1.42911
	μ = 0.9	1.03835	1.03950	1.03846	1.03841	1.03797	1.42945	1.42950	1.42962	1.42943	1.42913
Linear	Y = 0.5	1.03843	1.04080	1.03921	1.03862	1.03835	1.42860	1.43067	1.42923	1.42890	1.42894
	Y = 0.6	1.03830	1.04064	1.03937	1.03873	1.03847	1.42852	1.43062	1.42940	1.42907	1.42868
	Y = 0.7	**1.03789**	**1.03848**	**1.03873**	**1.03853**	**1.03796**	**1.42836**	**1.43056**	**1.42920**	**1.42852**	**1.42862**
	Y = 0.8	1.03894	1.04064	1.03905	1.03859	1.03826	1.42845	1.43073	1.42940	1.42890	1.42879
	Y = 0.9	1.03810	1.04080	1.03910	1.03868	1.03831	1.42873	1.43078	1.42890	1.42899	1.42893
Logistic	λ = 0.5	1.03841	1.04064	1.03969	1.03857	1.03842	1.42840	1.43028	1.42962	1.42868	1.42873
	λ = 0.6	1.03810	1.04065	1.03985	1.03857	1.03842	1.42840	1.42995	1.42968	1.42879	1.42884
	λ = 0.7	1.03826	1.04016	1.03953	1.03857	1.03810	1.42842	1.43012	1.42945	1.42879	1.42851
	λ = 0.8	1.03857	1.04032	1.03968	1.03842	1.03826	1.42846	1.43017	1.42940	1.42867	1.42846
	λ = 0.9	**1.03746**	**1.04000**	**1.03913**	**1.03826**	**1.03810**	**1.42806**	**1.42965**	**1.42923**	**1.42838**	**1.42840**
Power	ω = 0.5	1.03751	1.04016	1.03863	1.03889	1.03825	1.42840	1.43045	1.42915	1.42897	1.42895
	ω = 0.6	**1.03732**	**1.03968**	**1.03857**	**1.03841**	**1.03794**	**1.42823**	**1.42993**	**1.42899**	**1.42895**	**1.42896**
	ω = 0.7	1.03735	1.04000	1.03861	1.03841	1.03797	1.42845	1.43006	1.42901	1.42900	1.42900
	ω = 0.8	1.03767	1.04000	1.03878	1.03810	1.03778	1.42827	1.43006	1.42910	1.42903	1.42902
	ω = 0.9	1.03763	1.04016	1.03889	1.03856	1.03810	1.42835	1.43023	1.42912	1.42901	1.42901

The best entries are highlighted in bold

Hence it is recommended as it also outperforms the iGJ similarity measure without the time function.

6 Conclusion and future scope

To overcome the sparsity issue, a novel CF-based method iGJ is proposed in this paper. It outperforms other algorithms, i.e., Rating_Jaccard, RJaccard, RJMSD, Cosine, TMJ, etc. Along with this, time decay functions also integrated with the iGJ algorithm at the three levels of recommendation process. The experimental results indicate that the convex function provides best results at the similarity computation level and prediction level. In contrast, the power function produces good results at the rating matrix level. In all three, the result at the rating matrix level is much better than the other two approaches. So, through this paper, we can state that the results of the time based approach is better than the non-time-based approach. When a time function is applied with iGJ at similarity computation and prediction level, the convex function gives the best results in the time-based approach. At the rating matrix level, power function depicts satisfactory results. In comparison to these three, iGJ works efficiently when implemented with the power

Table 7 Results of applying TDF at SC level on Amazon magazine subscription dataset

Dataset	Parameters	MAE					RMSE				
		20	60	100	150	200	20	60	100	150	200
Concave	α = 0.5	1.24401	1.24451	1.24461	1.24487	1.24471	1.70472	1.70401	1.70472	1.70461	1.70474
	α = 0.6	1.24419	1.24467	1.24432	1.24478	1.24463	1.70489	1.70490	1.70489	1.70437	1.70489
	α = 0.7	1.24410	1.24464	1.24454	1.24463	1.24459	1.70457	1.70436	1.70454	1.70422	1.70440
	α = 0.8	1.24416	1.24432	1.24414	1.24464	1.24464	1.70496	1.70427	1.70468	1.70429	1.70393
	α = 0.9	**1.24348**	**1.24417**	**1.24385**	**1.24451**	**1.24409**	**1.70312**	**1.70362**	**1.70321**	**1.70355**	**1.70346**
Convex	β = 0.5	**1.24416**	**1.24406**	**1.24410**	**1.24412**	**1.24423**	**1.70312**	**1.70362**	**1.70360**	**1.70365**	**1.70372**
	β = 0.6	1.24459	1.24475	1.24445	1.24459	1.24459	1.70346	1.70412	1.70371	1.70424	1.70434
	β = 0.7	1.24501	1.24531	1.24497	1.24500	1.24501	1.70351	1.70393	1.70367	1.70412	1.70391
	β = 0.8	1.24496	1.24465	1.24475	1.24478	1.24490	1.70396	1.70397	1.70379	1.70431	1.70421
	β = 0.9	1.24497	1.24471	1.24473	1.24478	1.24491	1.70368	1.70383	1.70387	1.70436	1.70410
Exponential	μ = 0.5	**1.24416**	**1.24431**	**1.24432**	**1.24467**	**1.24436**	**1.70351**	**1.70355**	**1.70347**	**1.70359**	**1.70344**
	μ = 0.6	1.24510	1.24497	1.24495	1.24510	1.24481	1.70372	1.70369	1.70372	1.70372	1.70373
	μ = 0.7	1.24497	1.24485	1.24473	1.24491	1.24457	1.70387	1.70373	1.70396	1.70384	1.70386
	μ = 0.8	1.24520	1.24502	1.24485	1.24485	1.24469	1.70391	1.70382	1.70385	1.70361	1.70372
	μ = 0.9	1.24478	1.24490	1.24489	1.24499	1.24472	1.70356	1.70365	1.70378	1.70377	1.70378
Linear	Y = 0.5	1.24466	1.24471	1.24449	1.24456	1.24455	1.70356	1.70363	1.70370	1.70369	1.70381
	Y = 0.6	1.24465	1.24474	1.24473	1.24471	1.24440	1.70351	1.70362	1.70367	1.70368	1.70377
	Y = 0.7	**1.24456**	**1.24463**	**1.24447**	**1.24448**	**1.24438**	**1.70345**	**1.70355**	**1.70362**	**1.70366**	**1.70373**
	Y = 0.8	1.24492	1.24492	1.24451	1.24455	1.24447	1.70347	1.70359	1.70369	1.70372	1.70383
	Y = 0.9	1.24499	1.24499	1.24453	1.24468	1.24457	1.70353	1.70363	1.70371	1.70387	1.70389
Logistic	λ = 0.5	1.24451	1.24485	1.24501	1.24495	1.24499	1.70347	1.70392	1.70396	1.70397	1.70401
	λ = 0.6	1.24465	1.24476	1.24477	1.24482	1.24487	1.70355	1.70362	1.70364	1.70381	1.70382
	λ = 0.7	1.24462	1.24499	1.24481	1.24477	1.24463	1.70356	1.70371	1.70375	1.70372	1.70377
	λ = 0.8	1.24459	1.24463	1.24473	1.24469	1.24467	1.70353	1.70363	1.70371	1.70374	1.70382
	λ = 0.9	**1.24442**	**1.24445**	**1.24451**	**1.24453**	**1.24454**	**1.70342**	**1.70351**	**1.70353**	**1.70357**	**1.70358**
Power	ω = 0.5	1.24489	1.24468	1.24493	1.24469	1.24465	1.70276	1.70277	1.70287	1.70345	1.70327
	ω = 0.6	**1.24476**	**1.24461**	**1.24451**	**1.24448**	**1.24461**	**1.70191**	**1.70192**	**1.70213**	**1.70276**	**1.70290**
	ω = 0.7	1.24478	1.24467	1.24461	1.24481	1.24472	1.70224	1.70373	1.70243	1.70333	1.70326
	ω = 0.8	1.24484	1.24480	1.24480	1.24491	1.24476	1.70243	1.70338	1.70263	1.70338	1.70341
	ω = 0.9	1.24491	1.24489	1.24496	1.24466	1.24484	1.70259	1.70214	1.70277	1.70344	1.70339

The best entries are highlighted in bold

Table 8 Best value of tuning parameters

Function	Parameter	Best value @
Concave	α	0.9
Convex	β	0.5
Exponential	μ	0.5
Linear	y	0.7
Logistic	λ	0.9
Power	ω	0.6

function at the rating matrix levels of the recommendation process. It performs even better than iGJ without time function. Therefore, this work concludes that the performance of the iGJ with the power decay function at the rating matrix level is better.

Most of the existing CF-based recommender systems cannot scale up in a real-time environment. Thus, one of the future scopes of work will be the real-time implementation of the RS based on the iGJ with the time decay function. Future work also includes the deployment of the proposed system using contextual information.

Table 9 Results of applying TDF at RM level

Dataset	Functions	MAE					RMSE				
		20	60	100	150	200	20	60	100	150	200
ML-100k	Concave	0.51090	0.50960	0.51015	0.51105	0.51205	0.77582	0.77424	0.77508	0.77572	0.77650
	Convex	0.58665	0.58698	0.58825	0.59013	0.59113	0.87989	0.87808	0.87923	0.88032	0.88146
	Exponential	0.59738	0.59855	0.60043	0.60213	0.60330	0.85283	0.85349	0.85558	0.85715	0.85845
	Linear	0.47853	0.47813	0.47852	0.47883	0.47941	0.76328	0.76291	0.76370	0.76413	0.76470
	Logistic	0.46620	0.46618	0.46643	0.46745	0.46808	0.71603	0.71591	0.71598	0.71711	0.71776
	Power	**0.37615**	**0.37490**	**0.37568**	**0.37618**	**0.37703**	**0.67056**	**0.67000**	**0.67043**	**0.67121**	**0.67215**
Epinions	Concave	0.60223	0.60270	0.60270	0.60286	0.60286	0.93332	0.93391	0.93408	0.93417	0.93417
	Convex	0.79556	0.79477	0.79429	0.79413	0.79413	1.17918	1.17898	1.17864	1.17857	1.17857
	Exponential	0.81842	0.81683	0.81698	0.81714	0.81714	1.14781	1.14546	1.14580	1.14587	1.14587
	Linear	0.59159	0.58953	0.58873	0.58905	0.58905	0.91764	0.91652	0.91608	0.91626	0.91626
	Logistic	0.64508	0.64381	0.64381	0.64381	0.64381	0.91010	0.91026	0.91009	0.91032	0.91037
	Power	**0.40445**	**0.40429**	**0.40413**	**0.40397**	**0.40397**	**0.75488**	**0.75478**	**0.75482**	**0.75487**	**0.75486**
Amazon Magazine	Concave	1.06800	1.06816	1.06820	1.06817	1.06821	1.44077	1.44110	1.44103	1.44098	1.44102
	Convex	1.09822	1.09812	1.09817	1.09815	1.09828	1.45418	1.45415	1.45413	1.45419	1.45423
	Exponential	1.10876	1.10843	1.10852	1.10851	1.10872	1.51268	1.51292	1.51304	1.51311	1.51313
	Linear	1.00661	1.00636	1.00651	1.00661	1.00667	1.38144	1.38135	1.38146	1.38153	1.38157
	Logistic	0.80666	0.82123	0.83076	0.82456	0.82706	1.17268	1.18725	1.19378	1.19298	1.19178
	Power	**0.79532**	**0.79521**	**0.79544**	**0.79553**	**0.79566**	**1.13112**	**1.13107**	**1.13114**	**1.13134**	**1.13151**

The best entries are highlighted in bold

Table 10 Results of applying TDF at P level

Dataset	Functions	MAE					RMSE				
		20	60	100	150	200	20	60	100	150	200
ML-100k	Concave	0.76498	0.76815	0.77060	0.77245	0.77380	1.04689	1.04991	1.05207	1.05414	1.05513
	Convex	**0.75145**	**0.75470**	**0.75673**	**0.75975**	**0.76270**	**1.03714**	**1.03959**	**1.04093**	**1.04372**	**1.04623**
	Exponential	0.79933	0.79943	0.79950	0.79950	0.79945	1.07830	1.07871	1.07875	1.07912	1.07905
	Linear	0.76830	0.77063	0.77288	0.77523	0.77673	1.05003	1.05220	1.05450	1.05668	1.05767
	Logistic	0.75748	0.76080	0.76323	0.76583	0.76793	1.04081	1.04384	1.04550	1.04791	1.05010
	Power	0.77350	0.77605	0.77948	0.78105	0.78208	1.05380	1.05644	1.05940	1.06101	1.06197
Epinions	Concave	1.03889	1.03873	1.03762	1.03778	1.03778	1.42745	1.42718	1.42634	1.42640	1.42640
	Convex	**1.03714**	**1.03683**	**1.03619**	**1.03635**	**1.03635**	**1.42606**	**1.42595**	**1.42550**	**1.42556**	**1.42556**
	Exponential	1.04302	1.04143	1.04080	1.04096	1.04096	1.43167	1.43056	1.43012	1.43017	1.43017
	Linear	1.03889	1.03841	1.03730	1.03746	1.03746	1.42734	1.42706	1.42623	1.42629	1.42629
	Logistic	1.03825	1.03730	1.03667	1.03683	1.03683	1.42701	1.42634	1.42589	1.42595	1.42595
	Power	1.03937	1.03762	1.03730	1.03746	1.03746	1.42784	1.42689	1.42656	1.42662	1.42662
Amazon Magazine	Concave	1.25858	1.26190	1.26168	1.26206	1.26211	1.70914	1.71262	1.71256	1.71267	1.71281
	Convex	**1.24314**	**1.24646**	**1.24625**	**1.24672**	**1.24677**	**1.70443**	**1.70791**	**1.70785**	**1.70796**	**1.70810**
	Exponential	1.26099	1.26431	1.26410	1.26457	1.26462	1.71146	1.71494	1.71488	1.71499	1.71513
	Linear	1.25103	1.25435	1.25414	1.25461	1.25466	1.70614	1.70962	1.70956	1.70967	1.70981
	Logistic	1.24691	1.25003	1.25023	1.25029	1.25034	1.70574	1.70912	1.70910	1.70910	1.70921
	Power	1.25309	1.25641	1.25620	1.25667	1.25672	1.70674	1.71022	1.71016	1.71027	1.71041

The best entries are highlighted in bold

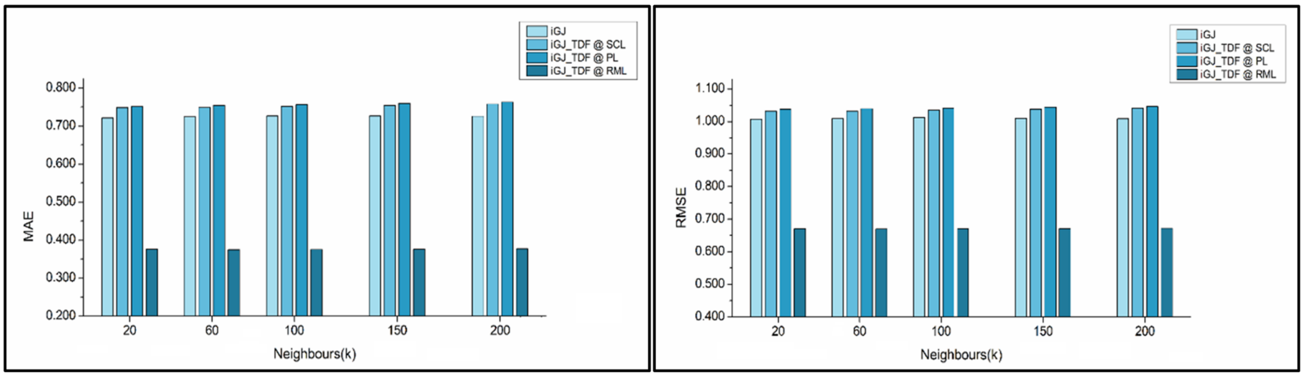

Fig. 4 iGJ without time function vs iGJ with time functions on ML-100k dataset

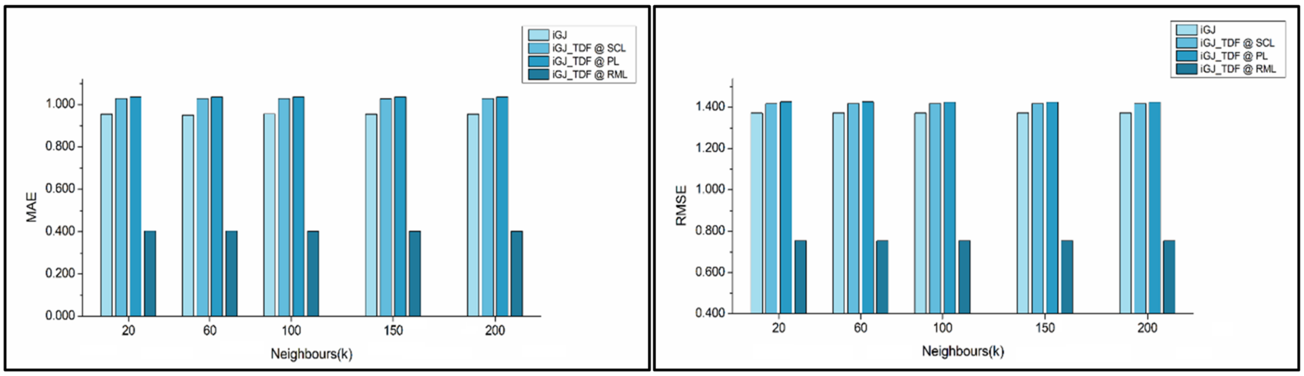

Fig. 5 iGJ without time function vs iGJ with time functions on Epinions dataset

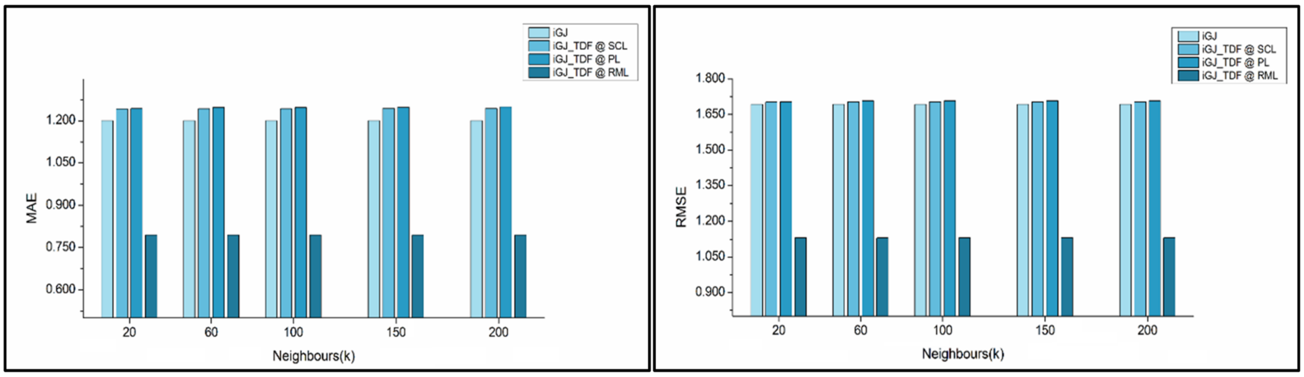

Fig. 6 iGJ without time function vs iGJ with time functions on Amazon magazine dataset

Funding No funding was received to assist with the preparation of this manuscript.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Adomavicius G, Mobasher B, Ricci F, Tuzhilin A (2011) Context-aware recommender systems. AI Mag 32:67–80. https://doi.org/10.1609/aimag.v32i3.2364

Ahn HJ (2008) A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf Sci 178:37–51. https://doi.org/10.1016/j.ins.2007.07.024

Al-bashiri H, Abdulgabber MA, Romli A, Hujainah F (2017) Collaborative filtering similarity measures: Revisiting. In: ACM international conference proceeding series part F1312, pp 195–200.https://doi.org/10.1145/3133264.3133299

Anandhan A, Shuib L, Ismail MA, Mujtaba G (2018) Social media recommender systems: review and open research issues. IEEE Access 6:15608–15628. https://doi.org/10.1109/ACCESS.2018.2810062

Ayub M, Ghazanfar MA, Khan T, Saleem A (2020) An effective MODEL for Jaccard coefficient to increase the performance of collaborative filtering. Arab J Sci Eng 45:9997–10017. https://doi.org/10.1007/s13369-020-04568-6

Ayub M, Ghazanfar MA, Mehmood Z et al (2019) Modeling user rating preference behavior to improve the performance of the collaborative filtering based recommender systems. PLoS ONE. https://doi.org/10.1371/journal.pone.0220129

Bag S, Kumar S, Tiwari M (2019) An efficient recommendation generation using relevant Jaccard similarity. Inf Sci. https://doi.org/10.1016/j.ins.2019.01.023

ben Ali B, Massmoudi Y (2013) K-means clustering based on gower similarity coefficient: a comparative study. In: 2013 5th international conference on modeling, simulation and applied optimization, ICMSAO 2013. https://doi.org/10.1109/ICMSAO.2013.6552669

Bulau J (2021) How much data is created every day in 2021? In: techjury. https://techjury.net/blog/how-much-data-is-created-every-day/#gref. Accessed 2 May 2022

Campos PG, Díez F, Cantador I (2014) Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model User-Adap Inter 24:67–119. https://doi.org/10.1007/s11257-012-9136-x

Chen YC, Hui L, Thaipisutikul T (2021) A collaborative filtering recommendation system with dynamic time decay. J Supercomput 77:244–262. https://doi.org/10.1007/s11227-020-03266-2

Ding Y, Li X (2005) Time weight collaborative filtering. In: International conference on information and knowledge management, proceedings, pp 485–492

Fontecha J, Hervás R, Bravo J (2014) Mobile services infrastructure for frailty diagnosis support based on Gower's similarity coefficient and treemaps. Mob Inf Syst 10:127–146. https://doi.org/10.3233/MIS-130174

Ghazanfar MA, Prugel-Bennett A (2010) A scalable, accurate hybrid recommender system. In: 3rd international conference on knowledge discovery and data mining, WKDD 2010 94–98. https://doi.org/10.1109/WKDD.2010.117

Ghazarian S, Nematbakhsh MA (2015) Enhancing memory-based collaborative filtering for group recommender systems. Expert Syst Appl 42:3801–3812. https://doi.org/10.1016/j.eswa.2014.11.042

Gong SJ, Cheng GH (2008) Mining user interest change for improving collaborative filtering. In: Proceedings—2008 2nd international symposium on intelligent information technology application, IITA 2008, vol 3, pp 24–27. https://doi.org/10.1109/IITA.2008.385

He L, Wu F (2009) A time-context-based collaborative filtering algorithm. In: 2009 IEEE international conference on granular computing, GRC 2009, pp 209–213

Huang X, Song Z (2014) Clustering analysis on E-commerce transaction based on K-means clustering. J Netw 9:443–450. https://doi.org/10.4304/jnw.9.2.443-450

Isinkaye FO, Folajimi YO, Ojokoh BA (2015) Recommendation systems: principles, methods and evaluation. Egypt Inform J 16:261–273. https://doi.org/10.1016/j.eij.2015.06.005

Jain G, Mahara T (2019) An efficient similarity measure to alleviate the cold-start problem. In: 2019 15th international conference on information processing: Internet of Things, ICINPRO 2019—proceedings. IEEE

Jain G, Mahara T, Sharma SC (2021) A collaborative filtering-based recommendation system for preliminary detection of COVID-19. In: Advances in intelligent systems and computing, pp 27–40

Jain G, Mahara T, Tripathi KN (2020) A survey of similarity measures for collaborative filtering-based recommender system. In: Advances in intelligent systems and computing, pp 343–352

Jain G, Mishra N, Sharma S (2013) CRLRM: category based recommendation using linear regression model. In: Proceedings—2013 3rd international conference on advances in computing and communications, ICACC 2013, pp 17–20. https://doi.org/10.1109/ICACC.2013.11

Jin Q, Zhang Y, Cai W, Zhang Y (2020) A new similarity computing model of collaborative filtering. IEEE Access 8:17594–17604. https://doi.org/10.1109/ACCESS.2020.2965595

Kant S, Mahara T (2018) Merging user and item based collaborative filtering to alleviate data sparsity. Int J Syst Assur Eng Manag 9:173–179. https://doi.org/10.1007/s13198-016-0500-9

Konstan JA, Miller BN, Maltz D et al (1997) Applying collaborative filtering to Usenet news. Commun ACM 40:77–87. https://doi.org/10.1145/245108.245126

Koren Y (2009) Collaborative filtering with temporal dynamics. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 447–455. https://doi.org/10.1145/1557019.1557072

Larrain S, Trattner C, Parra D et al (2015) Good times bad times: a study on recency effects in collaborative filtering for social tagging. In: RecSys 2015—proceedings of the 9th ACM conference on recommender systems. Association for Computing Machinery, Inc, pp 269–272

Lee TQ, Park Y, Park YT (2008) A time-based approach to effective recommender systems using implicit feedback. Expert Syst Appl 34:3055–3062. https://doi.org/10.1016/j.eswa.2007.06.031

Li D, Cao P, Guo Y, Lei M (2013) Time weight update model based on the memory principle in collaborative filtering. J Comput (finl) 8:2763–2767. https://doi.org/10.4304/jcp.8.11.2763-2767

Liu H, Hu Z, Mian A et al (2014) A new user similarity model to improve the accuracy of collaborative filtering. Knowl-Based Syst 56:156–166. https://doi.org/10.1016/j.knosys.2013.11.006

Ma H, King I, Lyu MR (2007) Effective missing data prediction for collaborative filtering. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval—SIGIR '07, p 39. https://doi.org/10.1145/1277741.1277751

Ma T, Guo L, Tang M et al (2016) A collaborative filtering recommendation algorithm based on hierarchical structure and time awareness. IEICE Trans Inf Syst E99D:1512–1520. https://doi.org/10.1587/transinf.2015EDP7380

Manochandar S, Punniyamoorthy M (2020) A new user similarity measure in a new prediction model for collaborative filtering. Appl Intell. https://doi.org/10.1007/s10489-020-01811-3

Nguyen LV, Nguyen TH, Jung JJ (2021) Tourism recommender system based on cognitive similarity between cross-cultural users. In: Intelligent environments 2021: workshop proceedings of the 17th international conference on intelligent environments. IOS Press, pp 225–232

Ni J, Li J, McAuley J (2019) Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: EMNLP-IJCNLP 2019—2019 conference on empirical methods in natural language processing and 9th international joint conference on natural language processing, proceedings of the conference, pp 188–197. https://doi.org/10.18653/v1/d19-1018

Patra BK, Launonen R, Ollikainen V, Nandi S (2014) Exploiting Bhattacharyya similarity measure to diminish user cold-start problem in sparse data. In: Lecture notes in computer science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol 8777, pp 252–263.https://doi.org/10.1007/978-3-319-11812-3_22

Podani J (1999) Extending Gower's general coefficient of similarity to ordinal characters. Taxon 48:331–340

Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) vol 4321. LNCS, pp 291–324 https://doi.org/10.1007/978-3-540-72079-9_9

Senior A (2017) Pearson's and cosine correlation. In: International conference on trends in electronics and informatics ICEI 2017 calculating 05 June 20, pp 1000–1004

Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:1–19. https://doi.org/10.1155/2009/421425

Sun SB, Zhang ZH, Dong XL et al (2017) Integrating triangle and Jaccard similarities for recommendation. PLoS ONE. https://doi.org/10.1371/journal.pone.0183570

Suryakant, Mahara T (2016) A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment. Procedia Comput Sci 89:450–456. https://doi.org/10.1016/j.procs.2016.06.099

Tan Z, He L (2017) An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle. IEEE Access 5:27211–27228. https://doi.org/10.1109/ACCESS.2017.2778424

Wang J, Zhang Y (2013) Opportunity models for e-commerce recommendation: right product, right time. In: SIGIR 2013—proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 303–312

Wang Y, Deng J, Gao J, Zhang P (2017) A hybrid user similarity model for collaborative filtering. Inf Sci 418–419:102–118. https://doi.org/10.1016/j.ins.2017.08.008

Wu D, Yuan Z, Yu K, Pan H (2012) Temporal social tagging based collaborative filtering recommender for digital library, pp 199–208

Xia C, Jiang X, Liu S, et al (2010) Dynamic item-based recommendation algorithm with time decay. In: Proceedings - 2010 6th international conference on natural computation, ICNC 2010. IEEE Computer Society, pp 242–247

Xu G, Tang Z, Ma C et al (2019) A collaborative filtering recommendation algorithm based on user confidence and time context. J Electr Comput Eng. https://doi.org/10.1155/2019/7070487

Yu C, Huang L (2017) CluCF: a clustering CF algorithm to address data sparsity problem. SOCA 11:33–45. https://doi.org/10.1007/s11761-016-0191-8

Zhang L, Zhang Z, He J, Zhang Z (2019) Ur: a user-based collaborative filtering recommendation system based on trust mechanism and time weighting. In: Proceedings of the international conference on parallel and distributed systems—ICPADS. IEEE Computer Society, pp 69–76

Zheng N, Li Q (2011) A recommender system based on tag and time information for social tagging systems. Expert Syst Appl 38:4575–4587. https://doi.org/10.1016/j.eswa.2010.09.131

Zimdars A, Chickering DM, Meek C (2013) Using temporal data for making recommendations. https://doi.org/10.48550/arXiv.1301.2320

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.